

Release 8 v17 SPT tools - May 12th, 2017

This release contains the tools, manuals and examples for the Signal Processing Toolbox SPT, SPT2 and SPT2.5 implementations.

S32 Design Studio for e200

The new graphical editor allows visually creating SPT routines in a graphic window accelerating the development process.

The included examples are based on the SPT validation tests. The file `fp_spt_test.c` contains the SPT controller and the file `fp_spt_test.spt` contains the SPT commands. They're provided as guidance.

GreenHills integration example

The GHS example contains a customization file that adds the SPT assembler to the build system for that project :

`RaceRunner_SPT_GHS\src_filetype\custom_filetype_binary.bod`.

In order to use the customization the `SPT_DIR` macro needs to be defined to indicate where to find the SPT tools installation, the example uses an environment variable set in the Windows control panel.

The project's "Set Build Options..." already contains a `-I $(SPT_DIR)\inc` option under the "SPT assembler" tab.

WindRiver integration example

The WR example is makefile driven, make it the top level

`RaceRunner_SPT_WindRiver`.

objdump

Since the SPT command live in their own sections objdump needs to have the `-D` option specified to force disassembling the `.spt` section, the `-r` embeds the relocation information.

New feature :

- Objdump prints instructions in assembler ready encoding
- Instruction evt supports selecting from bit 0 to bit 7

Bug fixes :

- Fixed `spt2` and `spt2.5` *scp's re_im_coeff encodings*, `.coef_cmplx`, `.coef_im`, `.coef_re` are now 00, 10, and 11 rather than 00, 01, 10
- `.transpose_fwd/.copy_fwd`, `.transpose_bck/.copy_bck` aliases added to copy instruction
- Fixed `spt2` and `spt2.5` *rdx2/irdx2 encodings removing .fft16 from the repeat operand legal values*
- Fixed `spt2` and `spt2.5` *scp.shift_wr, scp.shift_wr.ind shift_src encodings*
- In some cases the assembler couldn't report the usage of an undeclared identifier, the linker replaced that missing definition by 0.

- All tools, fixed a framework problem that caused a crash with wrong order of enumerated labels.
- SPT2.5, fixed an issue with the loop immediate instruction
- SPT2 and SPT2.5 documentation updates for cmp and rdx4 tw_ovs
- SPT2 and SPT2.5, added shift prescaling support to RDX2 and IRDX2:
`rdx2.shift_wr, rdx2.shift_wr.ind, rdx2.adptv.shift_wr,`
`rdx2.adptv.shift_wr.ind, irdx2.shift_wr, irdx2.shift_wr.ind,`
`irdx2.adptv.shift_wr, irdx2.adptv.shift_wr.ind,`
- Add sp2 and spt2.5 `rdx4.adptv.shift, rdx4.adptv.shift.ind,`
`rdx4.adptv.shift_wr, rdx4.adptv.shift_wr.ind`
- corrected the SPT2 instruction copy and copy.ind "blk_src_inc,
`blk_dest_inc, src_add_inc, dest_add_inc`" operand ordering
- added the SPT1 QuickRef mnemonic manual
- corrected the SPT1 manual for rd4x listing mult_mode and adptv
- corrected the SPT2 manual for pdma .cp6 and .cp4dfmtb fields
- corrected the SPT1 instruction PDMA data packing fields, replaced
`".16pack", ".24cmplx", ".24real", ".16real", ".48bin", ".16cmplx"`
with the complete and more coherent values
`".16cmplx", ".24cmplx", ".24real", ".16real", ".48bin", ".16swap",`
`".16clubbing"`
- inversion of PDMA's async and sync values
- support for R_PPC_ADDR32 relocations for sysram references in PDMA
- SPT2 on Windows didn't accept 64 bits operands
- SPT and SPT2 ADD destination WR was wrong if different from src1 WR